

## Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

## Listing of Claims:

1. (currently amended) A Virtual Desktop ~~—Meta-Organization. A Virtual Desktop—Meta-Organization & Control System~~, for use in a computer-processing environment having therein at least one processing unit with a respective operating-system, and the Virtual Desktop system comprises:
  - A. ~~In a real time accessible memory media~~, at least one dynamic substantially cyclic electronic-data structure;
  - [[B.]] ~~Associated~~ associated with each said data structure, an ongoing algorithmic activity that is respectively regularly
    - I. ~~Based on respective operating system data access, Transforming each transforming at least one process[[,]] [[of]] from~~ a plurality of processes executing running in the environment ~~or in a predetermined portion thereof, into~~ to an associated graphic representation; [[, and]]
    - [[II.]] ~~Logically~~ logically assigning the representation to a location in the data structure; and
  - C. ~~Associated with each said data structure~~, a graphic user interface facilitating displaying
    - [[I. On]] on a display device, ~~viewing of~~ the representations assigned to at least ~~one of the data structures or to~~ a portion of the data structure thereof, and
    - H. ~~Organizing of the at least one data structure in response to data received from a user, modifying the assignment of representations to locations in the data structure.~~
2. (currently amended) A Virtual Desktop ~~—Meta-Organization & Control System~~ according to claim 1 wherein the at least one dynamic substantially cyclic electronic-data structure includes a reduced resolution meta-data-structure ~~having~~ storing pointers to locations in the cyclic electronic-data structure.
3. (currently amended) A Virtual Desktop ~~—Meta-Organization & Control System~~ according to claim 1 wherein the ongoing algorithmic activity includes at least one program substantially as hereinbefore described and illustrated and selected from the list: UIManager (UI), MapManager

(MAP), AnimatorManager (ANIM), SystemHookManager (SYSHOOK), ScrollManager (SCROLLER), Executable Code Core Algorithm Group (ECCAG).

4. (currently amended) A Virtual Desktop ~~—Meta-Organization & Control System~~ according to claim 1 wherein the graphic user interface includes at least one program function substantially as hereinbefore described and illustrated and selected from the list: Window Grouping, 3D support, Sticky window, Multiple monitors support, Loop Compactification, Multiple Loop support, Increased MAX window size.
5. (currently amended) A Virtual Desktop ~~—Meta-Organization & Control System~~ according to claim 1 wherein the associated graphic representation is selected from the group consisting of a list: high resolution snapshot of a GUI of the process, a low resolution snapshot of a GUI of the process, a symbolic graphic representation for the process, a high resolution data stream of a GUI of the process, a low resolution data stream of a GUI of the process, a symbolic graphic representation data stream for states of the process.
6. (currently amended) A Virtual Desktop ~~—Meta-Organization & Control System~~ according to claim 1 wherein the plurality of processes is selected from the group consisting of a list:
  - A. at least two programs selected from the group~~[[:]]~~ consisting of electronic mail, word processing, streaming media, net-radio, net-television, net-video, web browser, chat room, electronic messaging, graphic application package, PowerPoint, Architecture support program, interior design support program, CAD/CAM, accounting support program, spread-sheet program;
  - B. at least two programs selected from the group~~[[:]]~~ consisting of real time financial data stream presentation program, transaction events validation program, aggregate analysis of transaction events program, collective transaction management support program, financial analysis alert program, financial analysis alarm program, day-trader interaction program, brokerage management directive program;
  - C. at least two programs selected from the group~~[[:]]~~ consisting of project management program, supply chain program, scheduling program, accounting program, project coordination program, resource allocation program;
  - D. at least two programs selected from the group~~[[:]]~~ consisting of ECG monitor program, EEG monitor program, physiological monitor program, medical history report program, drug interaction program, medical expert system program,

- correlation of physiological monitors program, medical condition alerts program, medical condition alarm program, medical information system program;
- E. at least two programs selected from the group[[:]] consisting of genomic data base series display program, local search genomic fragment identification computation program, correlation to known organic compounds identification program, genomic computation strategy comparison program;
- F. at least two programs selected from the group[[:]] consisting of artistic composition arrangement protocol, orchestration program cinematography production management program, animation program audio special effects program, visual special effects program, multimedia performance event program, film editing program, audio editing program, audio mixing program, visual series mixing and sequencing program;
- G. at least two programs selected from the group[[:]] consisting of an interactive command-control facility program, an observation monitoring program, a passive viewing of status program, an alert activating program, an alarm activating program; and
- H. a first program selected from any of the aforesaid groups, a second program selected from any of the aforesaid groups, and a third program interrelating data content from the first program with data content from the second program.
7. (canceled).
8. (new) Virtual Desktop according to claim 1, further comprising storing said data structure in non-volatile memory and further comprising retrieving said data structure from said non-volatile memory and initiating said processes logically assigned to locations in the data structure by reference to the transformed graphical representation stored in said data structure.
9. (new) A method of recalling an arrangement of program instances, comprising:
- providing a user interface for adding at least one instance of a program to a data structure storing program instance indicators, each program including a graphical interface which is indicative of a state of said program instance, the data structure storing said instances has an ordered closed loop linked group;
  - associating each of said added instances with a graphical representation;
  - displaying a plurality of representations from the graphical representations associated with instances added to said data structure in a horizontal linked orientation whereby adjacent graphical representations share at least one image frame border;
  - providing an interface for a user to horizontally scroll said displayed representations to reveal additional representations associated with reciprocal representations by said closed loop linked group;

providing an interface for a user to recall an instance of a program corresponding to a displayed graphical representation by selecting said graphical representation from said display;  
storing said data structure in non-volatile computer memory;

recalling said arrangement of program instances by retrieving said stored data structure for said non-volatile memory and displaying a plurality of the representations from graphical representations associated with instances stored in said data structure to allow the user to recall program instances.

10. (new) The method of Claim 9, further comprising providing a user interface for repositioning graphical representations within said horizontal display linked and updating data associated with at least one instant in said data structure in response to a user repositioning of a graphical representation associated with said instant.
11. (new) The method of Claim 9, further comprising periodically updating the program instance indicator for each instance in said data structure by reference to a current program instance state and updating said graphical representation for each said program instance.
12. (new) The method of Claim 9, further comprising in response to retrieving of said data structure initiating an instance of each program associated with an instance stored in said data structure, the initiating is of a state of the instance corresponding to the stored instance indicator in the data structure so as to have said instances available for providing to the user in response to user selection of graphical representations from said horizontal linked display.
13. (new) A method for managing application instances on a computer system, comprising:
  - generating a bitmap representation for each of a plurality of application instances executing on a computer system;
  - associating the bitmap representations with relative positions by storing identifiers for each representation and corresponding application instance with a corresponding position in a data structure to provide an ordered bitmap set;
  - displaying at least one application instance as a first active application instance, said displaying is substantially over a lower portion of the computer display;
  - displaying the bitmap representation associated with the current active application instance and at least two other bitmap representations along substantially a top portion of said computer display, the bitmap representations arranged linearly, extending horizontally and arranged by reference to the relative positions stored in said data structure;
  - providing a user interface for requesting a horizontal displacement of said displayed bitmap representations;
  - displaying at least one additional bitmap representation along said top portion in response to a user request for horizontal displacement; and

modifying the displayed first active application instance in response to a user request for horizontal displacement, said second active application instances identified by reference to the relative positions of the first active application instance and the second active application instance.

14. (new) A method for managing application instances on a computer system, comprising:

generating a bitmap representation for each of a plurality of application instances executing on a computer system;

associating the bitmap representations with relative positions by storing identifiers for each representation and corresponding application instance with a corresponding position in a data structure to provide an ordered bitmap set;

displaying at least one application instance as a first active application instance over a substantially lower portion of the computer display, the first active application instance associated with a first relative position;

displaying an adjacent application instance as the current active application instance, said adjacent application instance associated with a second relative position, in response to a user selection of a horizontal navigation control provided in said lower portion, said adjacent application instance identified by reference to said first relative position, said second relative position and the direction of horizontal navigation selected by said user; and

displaying the bitmap representation associated with the current active application instance and at least two other bitmap representations along a top portion of said computer display, the bitmap representations arranged linearly, extending horizontally and arranged by reference to the relative positions stored in said data structure.

15. (new) The method of Claim 14, further comprising modifying the relative position of said bitmap representations in said data structure by reference to user dragging of bitmap representations within said top portion display area.

16. (new) A method for managing application on a computer system display, comprising:

generating a bitmap representation for each of a plurality of application instances executing on a computer system;

associating the bitmap representations with relative positions by storing identifiers for each representation and corresponding application instance with a corresponding position in a data structure to provide an ordered bitmap set;

associating each bitmap representation with a relative position by storing the identifiers in a second data structure;

recalling application instances associated with said first data structure in response to user selection of said first data structure;



recalling application instances associated with said second data structure in response to user selection of said second data structure;

displaying at least one application instance as a current active application instance, the current active application instance associated with a first relative position;

displaying the bitmap representation associated with the current active application instance and at least two other bitmap representations along a top portion of said computer display, the bitmap representations arranged linearly, extending horizontally and arranged by reference to the relative positions stored in said data structure.

17. (new) The method of Claim 16, further comprising viewing updated states of applications instances by displaying updated bitmap representations of the application instances while the active application instance is not included in the data structure of the strip.
18. (new) The method of Claim 16, further comprising facilitating drag and drop functions between the active application and the application instances associated with the bitmap representations by transferring dropped data to a corresponding application by reference to the data structure storing the applicable bitmap representation.
19. (new) A method of recalling an arrangement of program instances, comprising:
  - providing a user interface for adding an instance of a program to a data structure storing program instance indicators, each program including a graphical interface which is indicative of a state of said program instance, the data structure storing said instances has an ordered closed loop linked group, the data structure further storing a location and size corresponding to each program instance;
  - storing said data structure in non-volatile computer memory;
  - recalling said arrangement of program instances by retrieving said stored data structure for said non-volatile memory and displaying a plurality of the representations from graphical representations associated with instances stored in said data structure by reference to said stored location and size for each program instance.
20. (new) The method of Claim 19, wherein said user interface comprises a user selection monitoring process that facilitates user selection of displayed data indicative of program instances and user direction of the selected data to the data structure so as to add the selected program instances to the data structure and further wherein said method automatically initiates the selected program instances in response to said user direction of the selected program instances to the data structure.
21. (new) The method of Claim 20, wherein said data structure further stored default data for program instance location and size which is referenced when automatically initiating program instances in response to said user direction.
22. (new) The method of Claim 20, wherein the user interface further includes an interface for specifying the position of the selected data within the data structure.